Videojuegos

Curso de Diseño y Programación

Nº 18 5,99 euros



Las **funciones** de red

Uso de los **"Trackers"**

El **audio** del juego









Curso práctico de Videojuegos

AUTOR DE LA OBRA
Marcos Medina
DIRECCIÓN EDITORIAL
Eduardo Toribio
etoribio@iberprensa.com
COORDINACIÓN EDITORIAL
Eva-Margarita García
eva@iberprensa.com
DISEÑO Y MAQUETACIÓN
Antonio Gª Tomé
PRODUCCIÓN
Marisa Cogorro
SUSCRIPCIONES
Tel: 91 628 02 03
Fax: 91 628 09 35
suscripciones@iberprensa.com
FILMACIÓN: Fotpreim Duvial
IMPRESIÓN: Gráficas Don Bosco
DUPLICACIÓN CD.ROM: M.P.O.
DISTRIBUCIÓN
S.G.E.L.
Avda. Valdelaparra 29 (Pol. Ind.)
28108 Alcobendas (Madrid)
Tel.: 91 657 69 00

EDITA: Iberprensa
www.iberprensa.com



CONSEJERO
Carlos Peropadre

REDACCIÓN, PUBLICIDAD Y ADMINISTRACIÓN
C/ del Río Ter, 7 (Pol. Ind. "El Nogal")
28110 Algete (Madrid)
Tel.: 91 628 02 03
Fax: 91 628 09 35
(Añada 34 si llama desde fuera de España.)

DEPÓSITO LEGAL: M-35934-2002
ISBN: Coleccionable: 84 932417 2 5
Tomo 2: 84 932417 4 1
Obra Completa: 84 932417 5 X

Copyright 01/07/03
PRINTED IN SPAIN

NOTA IMPORTANTE:
Algunos programas incluidos en los CD de "Programación y Diseño de Videojuegos" son versiones completas, pero en otros casos se trata de versiones demo o trial, versiones de evaluación que Iberprensa quiere ofrecer a nuestros lectores. No se trata en ningún caso de las versiones comerciales de los programas, y las hemos incluido para dar al lector la oportunidad de conocer y probar esos programas y que así pueda decidir posteriormente si desea o no adquirir las versiones comerciales de cada uno.

Aprende divirtiéndote

Bienvenidos de nuevo a **Programación y Diseño de Videojuegos**, la primera obra coleccionable cuyo objetivo es formar al alumno en las principales técnicas relacionadas en el desarrollo completo de un videojuego.

A lo largo de la obra el lector está aprendiendo programación a nivel general y a nivel específico con ciertas herramientas y lenguajes, aprendiendo a trabajar con aplicaciones de retoque de imagen y también de diseño 3D y animación. Estamos descubriendo las aplicaciones profesionales más importantes de audio y conociendo la historia de lo que se denomina "la industria del videojuego", los últimos 20 años, los juegos que marcaron un avance, sus creadores y en general la evolución del videojuego.

Pero además, esta obra tiene un segundo objetivo, desarrollar y potenciar la creatividad del lector, nosotros a lo largo de las diferentes entregas pondremos las bases y tú pondrás tu ingenio, tu creatividad y tu capacidad de mejorar.

Nos encontramos a mitad de camino del viaje de 20 semanas que os proponemos, viaje articulado en 400 páginas y 20 CD-ROMs cuya finalidad es proporcionar las bases mínimas para después cada uno continuar su camino.

Recuerda que para alcanzar el éxito necesitas cumplir tres condiciones: que te gusten los juegos, poseer cierta dosis de creatividad y finalmente capacidad de estudio.

Una la cumples seguro.

sumario

341 Zona de desarrollo
En esta entrega implementaremos todo el sistema de audio del juego, pero antes realizaremos un estudio de qué sistema utilizar para obtener unos resultados adecuados.

345 Zona de gráficos
Vamos a fabricar los gráficos correspondientes a los indicadores de pantalla y los paneles de fondo que sirven de soporte para la tabla de récords, ayuda y otros mensajes del juego.

349 Zona de audio
Comenzamos en este número con una serie dedicada a los secuenciadores denominados "Trackers".

351 Blitz 3D
Empezamos una serie de dos números dedicados a las funciones de red así como al manejo de las DirectPlay para sistemas multijugador.

355 Tutorial
Continuando con nuestro vídeo de presentación del juego, terminaremos el montaje del apartado visual con el editor de Ulead MediaStudio.

357 Historia del videojuego
Terminamos con los simuladores deportivos viendo los más destacados de golf, tenis, así como de deportes extremos.

359 Cuestionario
Cada semana un pequeño test de autoevaluación, en el próximo número encontrarás las respuestas.

360 Contenido CD-ROM
Páginas dedicadas a la instalación y descripción del software que se adjunta con cada coleccionable.

18

PARA ENCUADERNAR LA OBRA:

- *Tapas del volumen 1 disponibles en Iberprensa. Pedidos por teléfono: 916280203.*
- ***Tapas del volumen 2 ya a la venta en quioscos.***
- ***Los suscriptores recibirán las tapas en su domicilio sin cargo alguno como obsequio de Iberprensa.***

SERVICIO TÉCNICO:
Para consultas, dudas técnicas y reclamaciones Iberprensa ofrece la siguiente dirección de correo electrónico: games@iberprensa.com

PETICIÓN DE NÚMEROS ATRASADOS:
El envío de números sueltos o atrasados se realizará contra reembolso del precio de venta al público más el coste de los gastos de envío. Pueden ser solicitados en el teléfono de atención al cliente 91 628 02 03

Música y efectos de sonido

En esta entrega implementaremos todo el sistema de audio del juego.

Pero antes realicemos un estudio de qué sistema utilizar para obtener unos resultados adecuados.

(■) ANÁLISIS

La utilización de músicas en el juego es realmente sencilla de implementar, sin embargo debemos tener en cuenta qué formato de audio vamos a utilizar para soportarla. Si usamos música en formato MIDI (.MID) o en cualquier formato de secuencia tracker como .SM3, .MOD, XM, etc. debemos utilizar la reproducción con "PlayMusic". Como sabemos, este comando carga una pieza de música y la reproduce. Los inconvenientes de utilizar este sistema aparecen cuando queremos que la música se vaya repitiendo durante el juego o en el menú, ya que "PlayMusic" solo reproduce el tema una sola vez. Por lo tanto, es necesario volver a utilizar esta instrucción si queremos que la música suene continuamente:

musica = PlayMusic ("musica1.mid")
BUCLE PARTIDA
If ChannelPlaying (musica) = False
musica = PlayMusic (musica)
EndIf
FIN BUCLE

1

Procedimiento para ejecutar una música continuamente en el bucle del juego.

```
...
musica = PlayMusic ("musica1.mid")
...
While Not KeyHit(1)
  If ChannelPlaying (musica) = False
   Musica  = PlayMusic ( musica )
  EndIf
  PARTIDA
Wend
```

No parece que el problema sea muy complicado de solucionar. No obstante, el uso de "PlayMusic" puede ocasionar otro tipo de inconvenientes, sobre todo si estamos moviendo gráficos en la pantalla, ya que este comando carga de nuevo el fichero cada vez que es llamado.

Así que es posible que suframos al ver cómo nuestros gráficos se ralentizan en cada carga. Realmente, no es un problema muy alarmante, sobre todo si tenemos en cuenta que el tamaño de estos ficheros no es excesivo. Aun así, el uso de este tipo de formatos de audio para la música no es muy común en juegos actuales por su baja calidad. Por lo tanto, lo descartamos en beneficio de otros formatos más adecuados como .MP3 u .OGG . En nuestro caso utilizaremos el formato .OGG ya que posee una calidad y compresión muy similar al .MP3. La principal diferencia radica básicamente en el hecho de que el formato .OGG es de libre uso y no hay que pagar ningún tipo de licencia por incluirlo en nuestros juegos. Blitz3D es capaz de cargar y reproducir este tipo de ficheros por medio de "LoadSound", "PlaySound" y "LoopSound". La ventaja principal de "LoadSound" es que, una vez cargado y asignado a

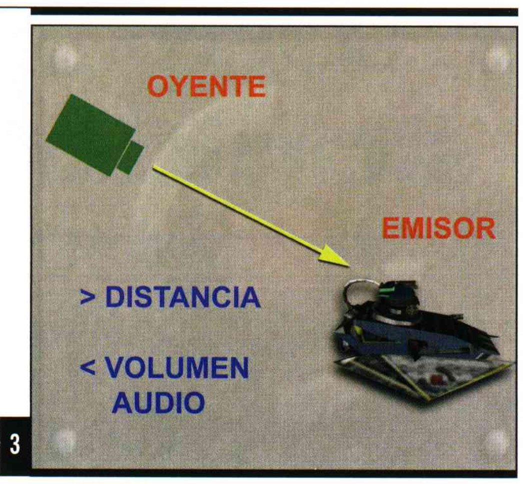
PlayMusic
.MID .MOD .SM3
. XM .WAV ...
PlaySound
.WAV .MP3 .OGG

2

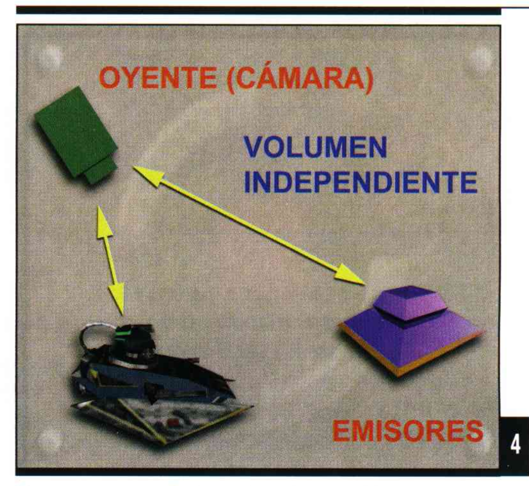
Formatos más comunes en el uso de "PlayMusic" y "PlaySound".

una variable, no es necesario volver a hacerlo cada vez que queramos reproducirlo. Esta ventaja se torna ideal para nuestros propósitos (Fig. 2).

Evidentemente, el desarrollo de un sistema de control de efectos de audio es crucial para que nuestro juego adquiera el matiz deseado. Podemos entonces pensar que el sistema tradicional de reproducción de audio no es el más conveniente si el juego se desarrolla en 3D, ya que perderíamos la sensación de estar dentro de un entorno. Debido a esto, presumiblemente podemos afirmar que la utilización de un sistema de audio 3D es el más idóneo. Como sabemos, en Blitz3D podemos trabajar con ambos sistemas, suficientes para resolver todas nuestras necesidades. Pero antes de decidirnos por uno u otro es preciso analizar cuáles son estas necesidades según el diseño previo del juego. El sistema 3D no es muy complicado de utilizar. Solo tenemos que asignar qué entidad será el oyente y cuál la que emite el sonido. Por ejemplo, para



El volumen del audio disminuye a medida que el emisor se aleja del oyente.



En nuestro procedimiento de audio se aplican volúmenes diferentes para cada emisor.

implementar el audio que genera una explosión de un disparo al chocar contra el terreno utilizaríamos:

```
...
Explosion = Load3Dsound ("explo.wav")
Oyente = CreateListener (camara)
...
Si disparo colisiona
  Reproduce_explosion (Disparo)
...
...
Function Reproduce_explosion (Emisor)
 EmitSound (Explosion, Emisor)
End Function
```

Debemos resaltar que es necesario cargar los sonidos utilizados para este sistema con la instrucción "Load3Dsound". Utilizar la entidad "camara" como oyente facilita la sensación de distancia entre emisores.

La desventaja de un sistema 3D de audio radica en el campo de acción donde trabaja. Dado que en nuestro juego las distancias son muy grandes, podríamos decir que este sistema no proporcionará el ambiente de combate que deseamos. La idea principal es que, independientemente de la posición de la bionave, el jugador pueda apreciar el sonido de la batalla más o menos destacado. Por lo tanto, nos armamos de valor y descartamos la facilidad que nos brindan las funciones de audio 3D de Blitz3D para fabricar nuestro propio sistema por medio de instrucciones tradicionales de reproducción.

(■) IMPLEMENTANDO LA MÚSICA

Una vez elegida la manera de utilizar la música en nuestro juego, pasamos a desarrollar el código necesario. En nuestro juego disponemos de música durante los menús y en cada partida. Para el primer momento utilizaremos una sola música, mientras que para las partidas jugaremos con dos diferentes elegidas al azar. Situándonos antes del bucle de cada partida en el módulo principal "zone_of_fighters.bb" escribimos el código de debajo (Ver Código 1).

Para evitar tener que cargar durante el juego la música de final de partida la tendremos almacenada previamente en la variable "musica2". Seguidamente, elegimos al azar entre los ficheros "musica4.ogg" y "musica5.ogg". Asignamos el canal necesario para la reproducción y activamos el volumen elegido desde el menú contenido en la variable global "Volumen_Musica". Durante el juego en la función "actualizar_juego()" del módulo "Fjuego.bb" realizamos la comprobación de fin de melodía y la reproducimos de nuevo:

```
If ChannelPlaying(c_musica1)=False
  c_musica1=PlaySound(musica1)
  ChannelVolume c_musica1,
  Volumen_Musica
EndIf
```

También debemos hacer esta misma comprobación en el bucle del menú principal y en el de los demás submenús.

(■) IMPLEMENTANDO EL SISTEMA DE EFECTOS DE SONIDO

Como analizábamos, vamos a desarrollar una función que permita simular un sistema 3D de audio por medio de instrucciones tradicionales de sonido 2D con la finalidad de tener las ventajas de ambos. Nuestro procedimiento solo controla la reproducción en dos altavoces estéreos, pero es suficiente para poder dotar a nuestro juego de un buen ambiente. Para realizar nuestra labor, aprovechamos las posibilida-

Código 1. Cargando la música

```
musica2=LoadSound("c:\zone of fighters\audio\musica3.ogg"); musica
del final
m=Rnd(4,5)
musica1=LoadSound("c:\zone of
fighters\audio\musica"+Str(m)+".ogg")
c_musica1=PlaySound(musica1)
ChannelVolume c_musica1,Volumen_Musica
```



5 *Para nuestra función de audio el volumen se mantiene estable cuando el oyente y el emisor coinciden.*



6 *En caso de que la cámara (oyente) se sitúe en tercera persona el volumen disminuye el doble.*

des que nos ofrece la utilización de canales para manejar el panorámico, por un lado, y la posibilidad de conocer la posición de cualquier entidad en el campo de visión de la cámara, por otro. Como sabemos, Blitz3D puede calcular la distancia entre dos entidades con la función "EntityDistance". Esta valiosa información nos servirá para asignar un volumen determinado a cada sonido dependiendo de la distancia a la cámara. Este procedimiento simulará la posición del emisor en el eje Y y Z. Para la simulación del eje X, es decir, determinar la posición lateral del emisor, utilizaremos la modificación del panorámico. Para averiguar la posición lateral de una entidad es necesario proyectar sus coordenadas en el entorno 3D a las coordenadas de la pantalla en 2D. Para ello, disponemos de las funciones "CameraProject" y "ProjectedX". La primera de ellas realiza la conversión, mientras que la segunda nos proporciona las coordenadas obtenidas. El entorno por donde nos movemos es de grandes proporciones, por lo tanto sería interesante reducir la reproducción de audio solo al campo de visión de la cámara. Blitz3D también nos ayuda con esta tarea mediante la función "EntityInView", la cual nos dice si una entidad es visible por la cámara. En caso de ser un *mesh*, la detección engloba también al radio de acción del sistema de colisiones asignado. Si por el contrario, la entidad es de otro tipo la referencia será su centro de posición.

Con todo este número de instrucciones es posible construir una función eficaz para nuestros propósitos. La implementamos en el módulo "functpantaudio.bb" y la función se llama "Play":

```
Function Play (sonido, canal,
emisor)
```

Los parámetros "sonido" y "canal" son utilizados por las instrucciones de audio y "emisor" proporciona la entidad que utiliza las funciones de proyección de coordenadas.

El procedimiento para llamar a la función lo podemos encontrar, por ejemplo, al realizar un disparo:

```
Play (sdisparo1,c_sdisparo1,
bionave)
```

En este caso, pasamos a la función la variable que contiene el sonido, el canal que lo soporta y la entidad "bionave" como emisor. Pasemos ahora a ver cómo utiliza la función estos parámetros para realizar una reproducción adecuada. Para simular mejor el efecto 3D utilizamos a la cámara como oyente. Así que calculamos en primer lugar su distancia al emisor del sonido con la función "EntityDistance" (Fig. 3 y 4):

```
Local distancia# =
EntityDistance (camara, emisor)
```

Con este valor almacenado en "distancia#" podemos utilizar varias sentencias para decidir qué volumen aplicar en cada caso (Ver Código 2).

Por ejemplo, en la cuarta sentencia condicional aplicamos un volumen de audio igual a la mitad del establecido en el menú del juego ("Volumen_SFX") si el emisor se encuentra entre 200 y 400 puntos de distancia de la cámara. Estas decisiones son realmente aproximadas y de su variación dependerá nuestro criterio y por medio del ensayo (Fig. 5, 6 y 7).

Una vez ajustado el volumen, el siguiente paso es reproducir el sonido por medio de "PlaySound" y con

Código 2. Decidimos qué volumen aplicar

```
Local volumen_audio#
If distancia#>10 And distancia<50      volumen_audio#=Volumen_SFX
If distancia#>50 And distancia<150     volumen_audio#=Volumen_SFX/1.5
If distancia#>500 And distancia<800    volumen_audio#=Volumen_SFX/2
If distancia#>200 And distancia<400    volumen_audio#=Volumen_SFX/2
If distancia#>800                      volumen_audio#=.1
```

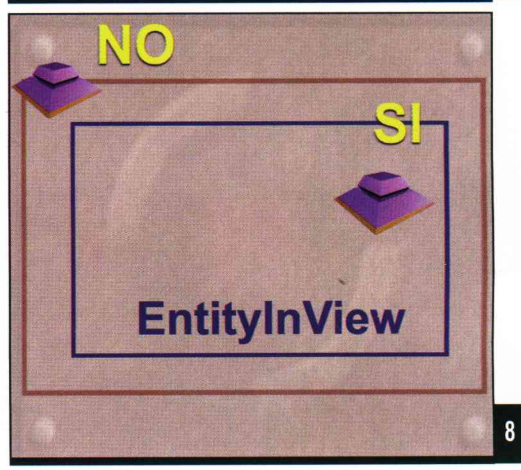


De igual forma, en una vista cenital el volumen se reduce a una cuarta parte.

visión de la cámara con "EntityInView" (Fig. 8):

```
If EntityInView
(emisor, camara) =
True
```

A continuación, dependiendo del valor de "ProjectedX", aplicamos un cambio de panorámico u otro. La idea es que dividamos el ancho de la pantalla en cuatro partes y para cada una aplicar una variación del estéreo con "ChannelPan" (Ver Código 3).



Esquema de la utilización de "EntityInView" para eliminar el audio de las entidades fuera del rango de visión de la cámara.

el volumen definido con "ChannelVolume":

```
Canal = PlaySound (sonido)
ChannelVolume canal,volumen_audio
```

Como sabemos, el panorámico debemos cambiarlo después de reproducir el sonido, así que pasamos a ello calculando la posición del emisor con respecto al oyente (cámara) con "CameraProject":

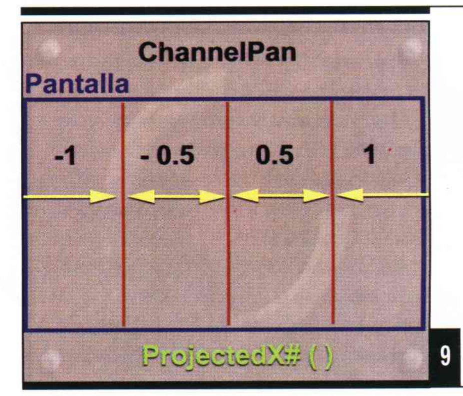
```
CameraProject (camara, EntityX
(emisor), EntityY (emisor),
EntityZ(emisor) )
```

Para obtener un efecto más acorde a la realidad, descartaremos todos los emisores que se encuentren fuera del rango de visión de la cámara con "EntityInView" (Fig. 8):

Se pueden realizar cuantas divisiones queramos para obtener unos resultados más finos, pero con cuatro es suficiente (Fig. 9).

Aun así, si es preferible efectuar un cambio en estas condicionales, utilizando valores dependiendo de la resolución de la pantalla en uso. Para ello, utilizamos las funciones "GfxModeWidth", que usamos para definir el modo gráfico ("mode") (Ver Código 4).

El sistema expuesto en este número permite, además, obtener un rendimiento muy óptimo cuando se reproducen al mismo tiempo gran cantidad de efectos sonoros, por ejemplo en plena batalla, sin ningún problema. Aunque siempre las limitaciones las pone la tarjeta de sonido que tengamos instalada en nuestro sistema.



Esquema de los valores que toma el panorámico dependiendo de la coordenada X almacenada en "ProjectedX()".

Código 3. Dividimos la pantalla en cuatro

```
If ProjectedX#()>1 And ProjectedX#()<100 ChannelPan canal,-1
If ProjectedX#()>100 And ProjectedX#()<200 ChannelPan canal,-.5
If ProjectedX#()>200 And ProjectedX#()<400 ChannelPan canal,.5
If ProjectedX#()>400  ChannelPan canal,1
```

En el próximo número...

... desarrollaremos todos los menús del juego.

Código 4. Definimos el modo gráfico

```
Local ancho= GfxModeWidth (mode)
If EntityInView(emisor,camara)=True
    If ProjectedX#()>1 And ProjectedX#()< ancho/6  ChannelPan canal,-1
    If ProjectedX#()> ancho/6  And ProjectedX#()<ancho/4 ChannelPan canal,-.5
    If ProjectedX#()>ancho/4 And ProjectedX#()<ancho/1.5 ChannelPan canal,.5
    If ProjectedX#()>ancho/1.5  ChannelPan canal,1
EndIf
```

Indicadores de pantalla y paneles

En esta entrega vamos a fabricar los gráficos correspondientes a los indicadores de pantalla y los paneles de fondo que sirven de soporte para la tabla de récords, ayuda y otros mensajes del juego.

La finalidad primordial que se busca en este número es conocer cómo trabajar con las herramientas para polígonos de Paint Shop Pro.

INDICADORES DE PANTALLA

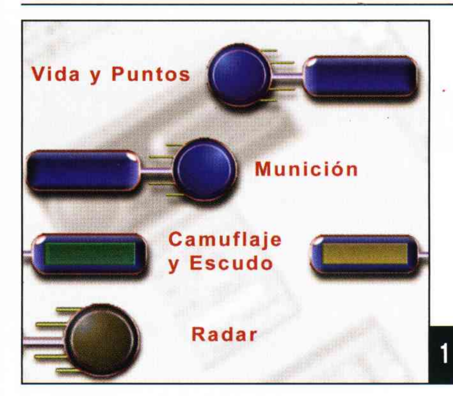
Según el diseño previo disponemos de siete indicadores para reflejar información sobre la puntuación, vida, munición, tiempo de escudo, de camuflaje y un radar (Fig. 1).

El diseño de todos ellos es el mismo, así que a partir de uno podemos realizar todos los demás. Elegimos el indicador de vida y puntos situado en la parte derecha de la pantalla. Para empezar, vamos a crear un documento nuevo con fondo negro de tamaño 210 x 70. El procedimiento que vamos a seguir es ir creando capas en las que añadiremos un nuevo elemento del dibujo. Empezamos por el "display" de los puntos. Creamos una capa nueva pulsando en el icono "Create Layer" de la paleta de capas. A esta capa la renombramos, por ejemplo, con el nombre "puntos".

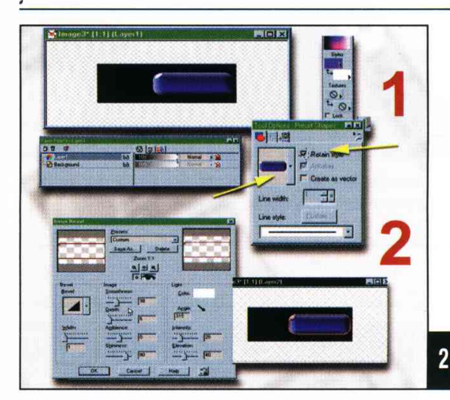
DISPLAY PARA PUNTOS

Elegimos la herramienta *Preset Shapes*. Aprovechamos los estilos que nos brinda la librería presente en la ventana *Tool Options*. Para poder utilizarlos es preciso activar la casilla *Retain style*. Abrimos la librería y elegimos el estilo "button008" (dejar el puntero sobre el dibujo del estilo para ver el nombre completo). Posteriormente, dibujamos un rectángulo como se muestra en la figura 2 / 1.

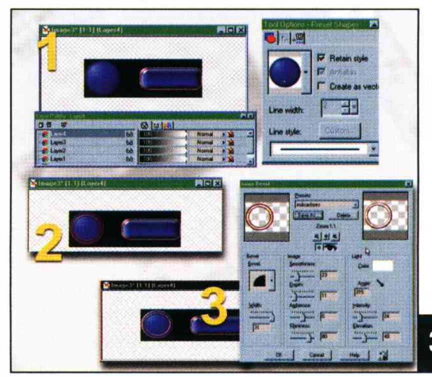
Observamos cómo adquiere el aspecto del estilo elegido que, como podemos ver, simula una pantalla de cristal. Para dar un toque más personal, añadimos un borde para dar la impresión de que la pantalla se encuentra dentro de un zócalo. Para ello, vamos a utilizar el mismo estilo pero esta vez con la casilla *Retain style* deseleccionada. En su lugar, activamos la casilla *Antialias*. Las demás opciones se harán visibles, así que elegimos un grosor de línea de 2 puntos en *Line witdh*. Creamos una nueva capa para dibujar el borde. Elegimos en la paleta de colores un tono granate para la tinta en *Styles*, activamos la nueva capa y en ella dibujamos, de nuevo, el mismo rectángulo. Observamos cómo solo se dibuja el contorno del estilo con el grosor elegido. Añadiremos un efecto 3D metálico para dar una mejor sensación de zócalo. Elegimos del menú *Effects* la opción *Inner Bevel* de la lista de efectos *3D Effects*. Vamos a crear un nuevo *preset* que nos servirá para todos los demás indicadores de este tipo. Seleccionamos la primera figura de la librería *Bevel* con un tamaño ("Width") de 1. En el apartado *Image* aplicamos los siguientes valores: Smoothness 38, Depth 5, Ambience 0 y Shininess 80.



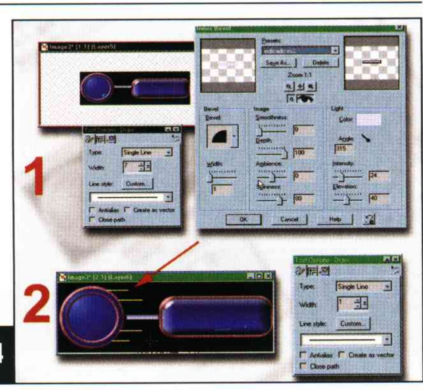
1 *Los diferentes tipos de indicadores para "Zone of Fighters".*



2 *La librería de estilos de la herramienta Preset Shapes es muy útil para obtener rápidamente buenos resultados.*

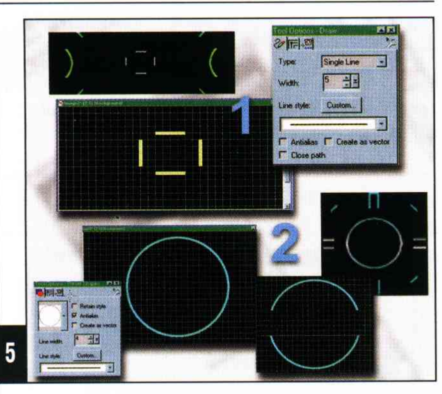


3 *Cambiando las opciones de la herramienta Preset Shapes podemos conseguir otro tipo de diseño poligonal para realizar bordes para una misma figura.*



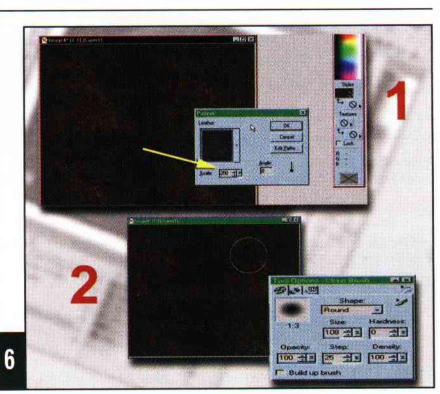
4

Añadir pequeños detalles es fácil y proporcionará un diseño más acabado y profesional.



5

Para realizar los puntos de mira es muy útil activar la cuadrícula para tener una mejor referencia.



6

Para evitar el tileado de las texturas se puede utilizar la herramienta de clonado y así eliminar partes iguales en el dibujo.

Y en el apartado *Light* seleccionamos un color de luz blanco, un ángulo de 315 grados y una intensidad (intensity) y elevación (elevation) de 25 y 40 respectivamente (Ver figura 2 / 2). Para finalizar, grabamos el nuevo *preset* (botón "Save as") con el nombre "indicadores".

DISPLAY PARA LA VIDA

El siguiente display del indicador es de forma circular y se utiliza para visualizar la vida del jugador. También debe parecer que es de cristal. De nuevo, aprovechemos la librería de estilos. Por lo tanto, la mejor opción es elegir el estilo "button015". No olvidemos tener activada la casilla *Retain style*. Creamos una nueva capa que llamamos "vida" y la activamos haciendo clic sobre ella. Nos situamos en la esquina superior izquierda de la pantalla. Para realizar un círculo perfecto, mantenemos pulsada la tecla "SHIFT" mientras desplazamos el ratón. Una vez creada la esfera, el siguiente paso es añadir el borde (Fig. 3 / 1).

Utilizando el mismo procedimiento anterior, seleccionamos un tono de color granate y desactivamos la casilla *Retain style*. Creamos una nueva capa y dibujamos el círculo (Fig. 3 / 2). A continuación, aplicamos el mismo efecto anterior para dar el aspecto metálico, pero esta vez con los parámetros cambiados. Seleccionamos la segunda figura en *Bevel* con el tamaño máximo (31). En el apartado *Image* seleccionamos: Smoothness 23, Depth 11, Ambience 0 y Shininess 80. En el apartado *Light* seleccionamos un color blanco, un ángulo de 315 grados y una intensidad y elevación de 24 y 40 respectivamente. Grabamos el *preset* con el nombre de "indicadores2" (Fig. 3 / 3). Añadiremos una línea con un tono celeste que servirá para unir los dos displays. Para realizarla utilizamos la herramienta de pincel (*Draw*) de tipo *Single Line* con un grosor de 5 puntos y la dibujamos en una nueva capa. Seguidamente, aplicamos el *preset* que grabamos anteriormente con el nombre "indicadores2" del efecto *Inner Bevel*. Una vez terminado, situamos la capa por debajo de las demás para disimular los extremos rectos de la línea (Fig. 4 / 1).

Para terminar este display añadiremos unas líneas de color amarillo que romperá con el estilismo del círculo. Partirán del extremo derecho hacia el display de puntos. Para su desarrollo utilizamos la herramienta del lápiz (*Draw*) con el tipo *Single Line* y con un grosor de 1 (Fig. 4 / 2). Una vez terminado el indicador lo salvamos en formato .PNG

El resto de indicadores tienen la misma forma. Por ejemplo, el indicador situado en el lado izquierdo correspondiente a la munición se obtiene aplicando un *Mirror* horizontal del menú *Image* a la imagen anterior grabada. También aplicando un cambio en el balance de color en *Color /Adjust/ Color Balance* obtendremos distintas versiones con diferente color.

PUNTOS DE MIRA

Otros indicadores que son necesarios dibujar son los puntos de mira. En "Zone of Fighters" disponemos de dos diferentes que aparecen cuando el jugador elige la cámara en primera persona. Uno para la munición de bajo calibre y misiles y otro para las bombas de minifusión y de retardo. Los puntos de mira, además de ayudar a realizar un disparo más certero, ayudan al jugador a saber qué tipo de arma ha seleccionado. También hay que tener en cuenta para su realización el utilizar formas estilizadas y tonos de color vistosos. Lo

primero sirve para evitar que se pierda atención en la acción ya que este gráfico se sitúa en el centro de la pantalla y lo segundo para que se pueda apreciar más claramente. Para realizarlos crearemos sus formas con trazos sueltos, pero que, en conjunto, se obtendrá la forma deseada. El primero consta de un cuadrado en el centro y dos semicírculos abiertos hacia fuera en los extremos y a cierta distancia. El cuadrado indicará la posición del objetivo y los semicírculos la separación de los cañones y por dónde pasarán los disparos. El segundo es básicamente un círculo dentro de otro. El más exterior está formado por secciones y el interior estará completo. No explicaremos cómo dibujar cada uno, sino cómo obtener polígonos segmentados.

Para realizar los cuadrados, simplemente activamos la cuadrícula con "CTRL" + "ALT" + "G" y con la herramienta del lápiz *Draw* dibujamos cuatro líneas separadas siguiendo las divisiones que se muestran en la figura 5 / 1.

Para los círculos segmentados, en primer lugar dibujamos el círculo con la herramienta para polígonos *Preset Shape*. Elegimos de la librería el primer estilo *Ellipse* y activamos la casilla *Antialias*. Seleccionamos el grosor de línea deseado y dibujamos el círculo (con la tecla "SHIFT" pulsada). Para obtener los segmentos, elegimos con la herramienta de selección rectangular el centro del círculo (incluyendo los bordes) y borramos, tal como podemos ver en la figura 5 / 2. Siempre debemos tener la cuadrícula visible para poder contar con una referencia y así obtener los segmentos iguales.

Hay que tener cuidado, ya que las líneas y curvas con suavizado o antialias generan bordes negros que luego se verán en el juego. Es necesario cuidar este detalle intentando utilizar lo menos posible el suavizado de bordes.

Otro detalle importante es utilizar el tamaño justo para los sprites y no desperdiciar espacio en blanco, ya que el tamaño influye posteriormente en el rendimiento del juego.

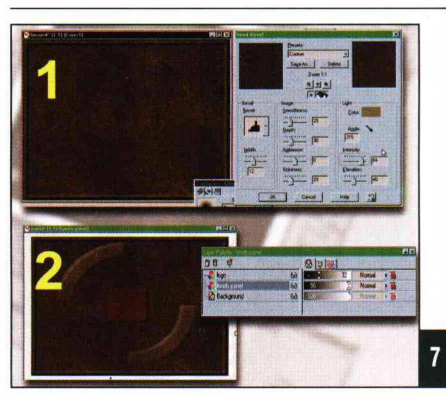
FABRICANDO LOS PANELES

Los paneles pueden tener diferentes tamaños dependiendo de para qué fondo serán utilizados. El panel estándar tiene aproximadamente un tamaño de 500 x 400. El aspecto que queremos darle es como si fuera un tablón de piedra clavado con cuatro tornillos. Para obtener el aspecto de la piedra pasamos a rellenar el fondo utilizando un patrón *Leather* elegido de la librería de estilos en *Styles*. En la ventana *Pattern*, donde elegimos los patrones, cambiamos el escalado de la textura en *Scale* a 200 para evitar al máximo el tileado (Fig. 6 / 1).

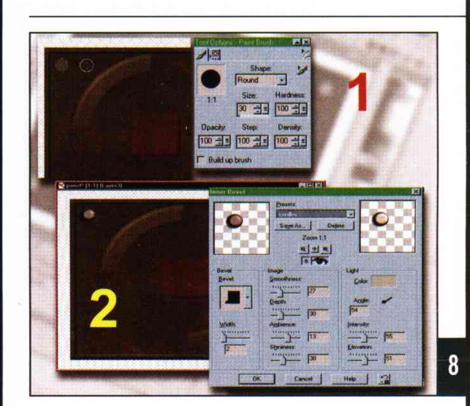
Seguidamente, rellenamos el fondo con la herramienta de relleno. De todas formas, todavía podemos observar cómo el tileado hace presencia. Para disimularlo vamos a utilizar la técnica del clonado de partes de la imagen para tapar aquellas zonas que se repiten. Elegimos la herramienta de clonado *Clone Brush*. En la ventana de opciones elegimos un tamaño de pincel considerable, más o menos de 100 unidades. El procedimiento a seguir es el siguiente:

En primer lugar, con la tecla "SHIFT" pulsada hacemos clic sobre la zona de la imagen que queremos copiar y seguidamente pintamos sobre la zona donde queremos que la parte seleccionada se copie. Se irá clonando lo que se encuentra debajo del aspa (Fig. 6 / 2).

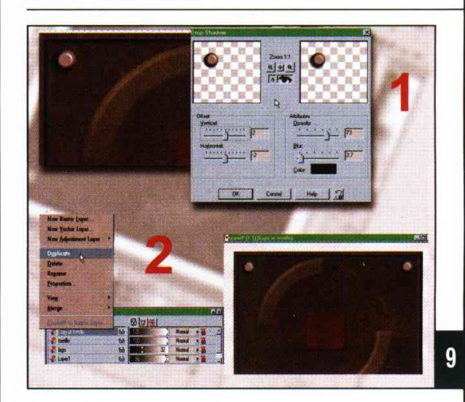
Una vez retocada la imagen, el siguiente paso es



Configurando correctamente el efecto 3D Inner Bevel se puede obtener multitud de efectos interesantes como los bordes del panel.



A partir de un simple punto de color plano podemos obtener la cabeza de un tornillo con volumen, gracias de nuevo al efecto Inner Bevel.



A partir del primer tornillo podemos obtener todos los demás duplicando las capas.



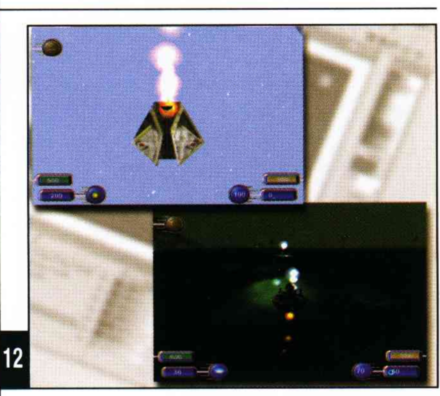
10

Dos ejemplos del uso de los paneles diseñados en esta entrega.



11

El uso de paneles para dar soporte a diferentes textos como la tabla de puntuación ayuda a obtener un aspecto más profesional al juego.



12

Nuestro diseño de indicadores es eficaz ya que proporciona multitud de información sin apenas ocupar espacio en la pantalla.

dibujar un borde escalonado para obtener la forma de tablón. Para realizarlo utilizamos de nuevo el efecto 3D *Inner Bevel* tan útil. En esta ocasión elegimos la figura sexta para el parámetro *Bevel* con un tamaño de 12. En el apartado *Image* los parámetros son: Smoothness 25, Depth 30, Ambience 0, y Shniness 29. En el apartado *Light* un tono marrón con un ángulo de 315 grados. La intensidad a 84 y la elevación a 40. Añadamos, mezclado con el fondo, el logotipo del fondo (Fig. 7 / 1).

Cargamos la imagen del logo (si no la tenemos a mano se puede encontrar en "Extras" del CD). Arrastramos la capa del fondo hacia el documento del panel. Habremos creado una nueva capa con la imagen del logo junto con el fondo de color blanco. Para eliminar este color blanco lo seleccionamos con la herramienta de selección *Magic Wand* con la opción de tolerancia al 25 % y luego borramos. Para realizar la mezcla cambiamos la opacidad de la capa al 30 o 35 % (Fig. 7 / 2).

El siguiente proceso es crear los tornillos que están clavados en el panel. Para realizarlos vamos a partir del dibujo de un círculo hecho con la herramienta de pincel. En sus opciones, seleccionamos una forma redonda (*Shape*) con un tamaño de 30 puntos y todos los demás parámetros al 100 %. Creamos una nueva capa que servirá para alojar el primer tornillo (Fig. 8 / 1).

Para obtener la forma de la cabeza del tornillo, de nuevo, utilizaremos un efecto 3D. Como siempre, el mejor resultado lo obtenemos aplicando un efecto *Inner Bevel*. Para obtener la forma elegimos la figura 5 en *Bevel* con 2 de tamaño. Los parámetros para el apartado *Image* son los siguientes: Smoothness 27, Depth 30, Ambience 13 y Shininess 38. Para el apartado de luz seleccionamos una tonalidad grisácea para el color con un ángulo de 54 grados. En la elección del ángulo de incidencia de la luz, tenemos que tener en cuenta que coincida con la luz que le llega al panel, la cual podemos observar en sus bordes. Terminamos, con una intensidad del 55% y una elevación del 51 (Fig. 8 / 2).

Debemos añadir también un efecto de sombreado para realzar más la altura de la cabeza del tornillo. Por lo tanto, seleccionamos del menú *Effects /3D Effects* el efecto *Drop Shadow* y le aplicamos un desplazamiento vertical (*Offset / Vertical*) de 2 y uno horizontal (Horizontal / Offset") de –2. En los atributos aplicamos una opacidad (*Opacity*) del 73%, un desenfoque (*Blur*) de 3,7 puntos y con color negro (Fig. 9 / 1).

Una vez acabado el tornillo, duplicamos la capa para obtener los otros tres (Fig. 9 / 2).

Para obtener el resto de paneles solo es necesario realizar operaciones de escalado a la capa del fondo del panel (la tabla) y después, solo tenemos que utilizar la herramienta de texto para disponer de los diferentes avisos.

Como siempre, el diseño de los indicadores de pantallas se debe acomodar al conjunto del juego y tener la estética del diseño global. Muy importante tener siempre en mente que esta parte del juego tiene que ser, ante todo, una útil y clara herramienta de comunicación con el jugador.



En el próximo número...

... seguiremos con otros elementos del juego como las texturas para los bonos o los diferentes tipos de sprites.

Secuenciadores trackers (I)

Comenzamos en este número con una serie dedicada a los secuenciadores denominados "Trackers".

Estos secuenciadores son una buena alternativa al desarrollo de música por ordenador si no se dispone de instrumentos externos. Combinan perfectamente muestras e instrumentos a través de patrones, los cuales unidos conforman la canción. Estos programas son fruto de la corriente "Demo Scene" de la época de los primeros Amiga y Atari de cuya herencia encontramos muchos de ellos de uso gratuito. Hemos elegido uno de los mejores "Trackers" gratuitos para Windows, ModPlug Tracker.

MODPLUG TRACKER

Como hemos comentado, la integración de este secuenciador en el entorno Windows lo hace más atractivo y cómodo que otros basados en el viejo DOS. Sin embargo, la ventaja de trabajar bajo Windows tiene su precio y son los retardos que a menudo podemos encontrar al reproducir los temas. Aun así, el uso del ratón y los atajos de teclado habituales en las aplicaciones bajo este sistema operativo convierten a ModPlug Tracker en una herramienta ideal. Sin más dilación pasemos a conocer su funcionamiento.

Para empezar a trabajar debemos crear un documento nuevo. ModPlug Tracker trabaja con cuatro tipo de formatos de canción: IT (Módulos Impulse Tracker), XM (FastTracker II), S3M (ScreamTracker) y MOD (ProTracker). En la opción *File / New* podemos elegir cualquiera de estos formatos. Pero lo que más sorprende de este programa es la cantidad de formatos diferentes que puede cargar. Incluso ficheros olvidados de los primeros años y módulos empaquetados (como .ZIP o .MDZ). Todos los ficheros cargados se convertirán a IT, XM, S3M o MOD para poder ser editados por el programa (Fig. 1).

Para empezar, la mejor opción es el formato .S3M. Sin embargo, vamos a elegir el formato IT que prácticamente es igual pero con la posibilidad de añadir instrumentos, ya que .S3M solo trabaja con patrones de muestras.

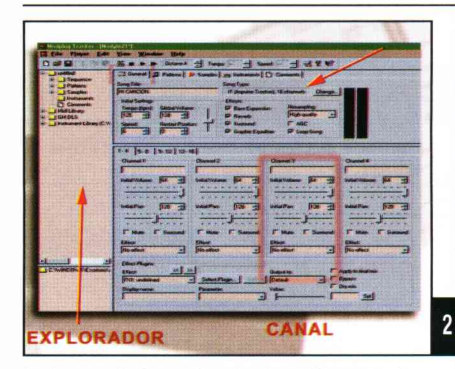
Al crear el nuevo documento en formato .IT, se abre la ventana de edición de la canción. Esta ventana está dividida en seis secciones conmutables a través de pestañas.

PANTALLA "GENERAL"

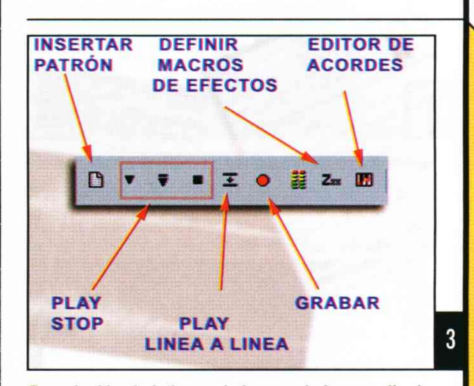
Es la primera de las secciones y es donde configuramos nuestra canción. Esta ventana está dividida en varias secciones. En la parte superior encontramos el apartado *Initial Settings* donde podemos seleccionar el "tempo" en negras por minuto, el volumen general de la canción (*Global Volume,* Máximo 128), su velocidad y la posición inicial de reproducción. El nombre de nuestra canción lo colocamos en *Song Title* y en *Song Type* el tipo de canción; es decir, que podemos cambiar nuestra canción a otro formato diferente o seleccionar nuevas compatibilidades para el formato que estamos utilizando. Por último en la sección *Effects* podemos seleccionar algunos efectos modificables en tiempo de ejecución. En la parte inferior de la pantalla se encuentra la sección dedicada a la configuración de cada canal disponible en grupos de cuatro. Desde



Lista de formatos que admite ModPlug Tracker.



En la pantalla General es donde configuramos la canción.



Descripción de la barra de iconos de la pantalla de patrones.

el nombre hasta el tipo de efecto que tendrá aplicado (Fig. 2).

PANTALLA "PATTERNS"

En esta sección es donde escribimos la canción. Funciona exactamente igual que los demás secuenciadores pero con la ventaja de que la navegación por ella se puede realizar cómodamente con el ratón. En la parte superior se encuentran todos los botones de transportes y de opciones. En las figuras 3 y 4 se muestra una descripción de cada uno de ellos.

Trabajaremos más adelante con cada una de estas opciones. Para elegir un instrumento lo hacemos en *Instrument*. La casilla *Spacing* es muy interesante y nos permite saltar filas en los patrones de notas a la hora de grabar, tantas como se indique. Ya practicaremos con su uso más adelante.

En la parte inferior de esta pantalla se encuentra la sección de patrones donde escribimos cada una de las notas de la canción. Podemos ver cada uno de los canales representados por columnas y divididos por una línea vertical. Cada columna se encuentra divida en filas enumeradas del 0 al 63. Cada una de estas filas lleva cuatro valores. La primera parte representa la nota en notación alfanumérica, la segunda representa el sonido de esa nota, la tercera el volumen y la cuarta el efecto asociado. Haciendo clic en cada una de estas cuatro partes aparecerá una ventana flotante donde podemos elegir cada uno de estos parámetros (Fig. 5).



NOTA

Una canción en ModPlug Tracker se forma a partir de la unión de patrones (patterns) y cada patrón está formado por 64 notas por defecto.



NOTA

Para navegar por los patrones se pueden utilizar también los cursores. Además, se pueden seleccionar con el ratón grupos de notas y editarlas con las funciones estándar de Windows.

PANTALLA "SAMPLES"

ModPlug Tracker no permite la grabación de muestras, solo la modificación de ficheros de audio adquiridos. En esta pantalla podemos realizar estas modificaciones. Desde esta pantalla cargamos todas las muestras que servirán de instrumentos para utilizar en los patrones. En la casilla *Sample* podemos elegir las muestras que vayamos cargando. Podemos observar que es posible modificar parámetros como la frecuencia, el volumen o el tono, así como la posibilidad de crear bucles (loops) y aplicar ciertos efectos. En la parte superior derecha encontramos algunos iconos de opciones que describimos en la figura 6.

PANTALLA "INSTRUMENTS"

En esta parte del programa podemos crear y modificar nuestros instrumentos. Por instrumentos se entiende también muestras en formato .WAV. Esta sección nos brinda la posibilidad de cambiar la envolvente del tono, volumen o panorámico a nuestro gusto a través de un editor gráfico situado en la parte inferior. Esta pantalla requiere un estudio especial que iniciaremos en la próxima entrega (Fig. 7).

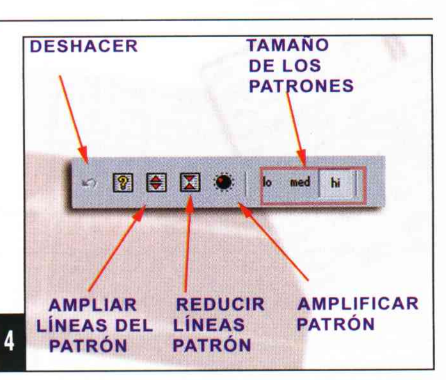
PANTALLA "COMMENTS"

A diferencia de otros "trackers" ModPlug proporciona la opción de poder describir e ilustrar con texto las canciones de manera bastante libre en la casilla *Song Message*.

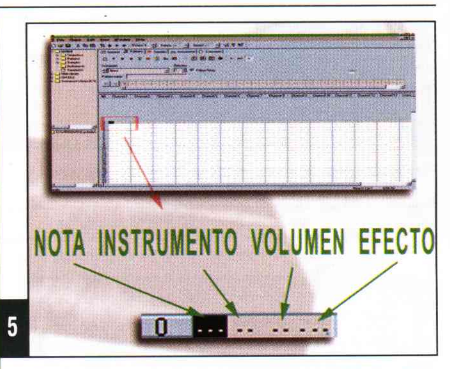


En el próximo número...

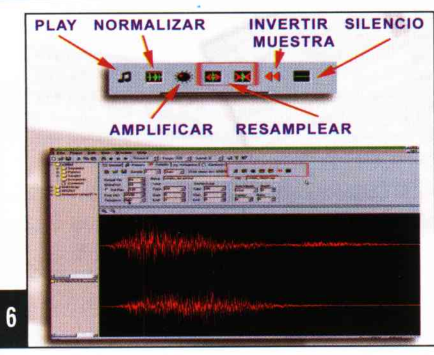
... entraremos de lleno en un ejemplo práctico.



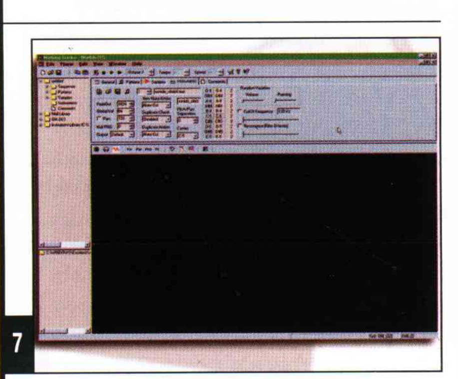
Descripción de la barra de iconos de la pantalla de patrones.



Elementos que forman cada línea de nota de cada patrón.



Descripción de los iconos de la pantalla Samples.



Ejemplo de envolvente aplicado a un instrumento.

Funciones de red y protocolos (I)

Con esta entrega empezamos una serie de dos números dedicados a las funciones de red así como al manejo de las DirectPlay para sistemas multijugador.

Pero antes debemos conocer algunos conceptos básicos necesarios para comprender el funcionamiento de estas funciones.

CONCEPTOS BÁSICOS

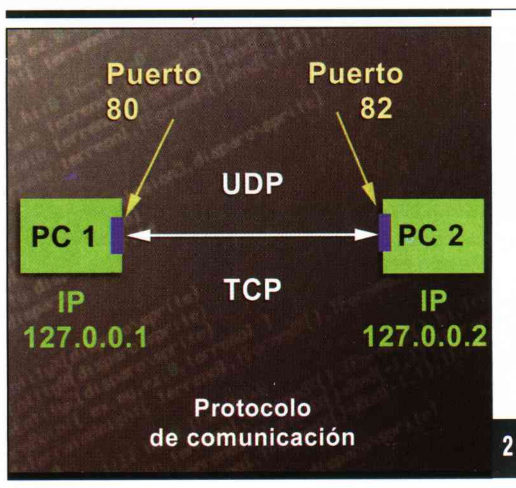
No es necesario un profundo conocimiento de los sistemas de comunicación entre ordenadores en sistemas de área local o a través de la red de redes (Internet). Sin embargo, sí es preciso conocer algunos conceptos utilizados por estos sistemas y que determinan la forma de trabajar de las funciones de red que Blitz3D posee. Con la utilización de esta cualidad que este lenguaje nos brinda entramos en un nuevo mundo de posibilidades presente en casi todos los juegos de hoy día. En un principio puede resultar algo complicado entender su funcionamiento pero realmente resultan gratificante los resultados que se pueden llegar a obtener.

COMUNICACIÓN ENTRE ORDENADORES

Básicamente la manera más sencilla de comunicación entre ordenadores es a través de una red de área local o LAN (Local Area Network). Este sistema puede constar de dos ordenadores conectados entre sí (un servidor y un cliente) o por cientos de ellos, los cuales pueden compartir archivos, dispositivos, etc. Este tipo de conexión es la más directa y rápida, y la pérdida de datos transmitidos es casi nula. Otra manera de conectar ordenadores entre sí es a través de Internet. Realmente, la comunicación por Internet es como si fuera una red local. La diferencia radica en que es de manera menos directa ya que utiliza una conexión telefónica. La ventaja es que se pueden conectar ordenadores situados en cualquier lugar del mundo.

Ahora bien, para comunicar dos ordenadores o varios entre sí, ya sea por área local o por internet, es necesario utilizar un protocolo especial para la transmisión de datos para indicar cómo se efectúa la transferencia. Los más usados son el protocolo TCP (Protocolo de Control de Transmisión) y UDP (Protocolo de Datagramas de Usuario) para área local y TCP/IP (Protocolo de Control de Transmisión / Protocolo de Internet) para Internet. Aunque, como veremos, es posible mezclarlos para obtener mejores resultados. Estos protocolos configuran un sistema de envío de información por paquetes de bytes al ordenador/es de destino. Explicaremos con más detalle cada uno de ellos en su debido momento.

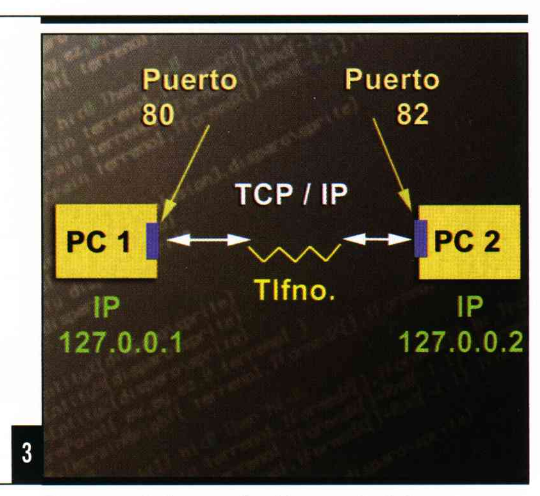
Es evidente que todos los ordenadores conectados entre sí necesitan estar identificados de alguna manera para poder compartir información. Para ello, se utilizan las direcciones IP (Internet Protocol). Cada dirección IP está formada por cuatro números enteros del 0 al 255 separados por un punto. Todas son exclusivas y no puede haber dos ordenadores con el mismo IP. Aunque es un protocolo propio de Internet se utiliza de



Elementos de la comunicación directa entre dos ordenadores.

1

Un ejemplo sencillo de Área Local (LAN) con cuatro ordenadores y un servidor.



Elementos de la comunicación a través de internet entre dos ordenadores.

identificada y clasificada. Por lo tanto, sí resulta ideal para juegos basados en turnos como juegos de mesa, puzzles o ciertos juegos de Estrategia y Rol donde no importa el tiempo de espera.

Por otro lado, la comunicación por UDP no garantiza la llegada de todos los paquetes de información ya que busca el camino más corto y de forma desordenada. Sin embargo, debido a esto, la velocidad de transmisión es mucho más elevada que TCP, por lo que es ideal para cierto tipo de juegos de acción en donde la actualización del juego se hace en tiempo real, con lo que la pérdida de determinados datos resulta irrelevante. Por otro lado, los paquetes de información son más difíciles de controlar y de saber su tipo y origen.



Significado y contenido de los puertos de comunicación y del protocolo de Internet.

igual manera para una conexión directa entre equipos.

SISTEMAS MULTIJUGADOR

Blitz3D también permite utilizar un sistema de comunicación propio para juegos a través de las DirectPlay de Microsoft. Con este sistema se pueden crear y gestionar partidas para varios ordenadores conectados directamente por cable o por Internet por medio del envío de mensajes perfectamente ordenados. Es bastante sencillo de utilizar ya que posee un gran número de funciones específicas y especializadas.

DIFERENCIAS ENTRE TCP Y UDP

Es muy común tener dudas a la hora de decidir qué sistema de comunicación utilizar para nuestros proyectos. Así que lo mejor será conocer cuáles son las diferencias más importantes de cada uno de ellos. En el protocolo TCP los paquetes de información llegan a un destino conocido de forma ordenada. Esta fiabilidad cuesta rendimiento y resulta más lento para su utilización en juegos que requieren datos continuamente en tiempo real. Sin embargo, su uso es sencillo ya que la información está continuamente

FUNCIONES PARA GESTIONAR EL PROTOCOLO TCP/IP

El protocolo TCP está orientado a la conexión entre ordenadores y el protocolo IP solo al uso de la red. TCP es extremadamente fiable en conjunción con IP por lo que son usados conjuntamente para la transmisión a través de Internet de datos correctamente ordenados.

Blitz3D puede utilizar este protocolo para conformar una comunicación entre ordenadores conectados a Internet (TCP/IP) o bien para ordenadores conectados directamente. Aunque generalmente se utiliza en aplicaciones para Internet como servidores FTP y juegos on-line.

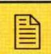
El concepto es realizar la comunicación entre servidor y cliente. Para asignar o crear un servidor a través de un determinado puerto de comunicación disponemos de la función "CreateTCPServer":

```
CreateTCPServer (Puerto de
comunicación)
```

DEFINICIÓN

▶ FTP

FTP es un protocolo de transferencia de ficheros que sirve para que un usuario pueda recibir o enviar ficheros a través de la red.



NOTA

Es evidente que si una comunicación TCP se hace entre ordenadores fuera del área local es necesario tener una conexión a Internet.



NOTA

Un puerto es por donde entran y salen los datos de la comunicación. Por lo tanto, es necesario asignar un puerto diferente para cada ordenador. Se trata de un número entero.

Por ejemplo, si queremos que nuestro ordenador sea el servidor:

```
Servidor = CreateTCPServer (2020)
If Servidor <> 0 Print
"Servidor creado ..."
```

Una vez creado el servidor podemos recibir o enviar información desde otros ordenadores (clientes). El funcionamiento es similar al uso de ficheros. Consideremos a cada sistema conectado como un disco duro en el que se graban o leen datos de forma lineal. En un sistema por red estos datos se almacenan secuencialmente en paquetes de información o "Stream". Cada tipo de protocolo maneja estos paquetes de diferente forma. Para abrir un "Stream" en el servidor y poder enviarle paquetes de datos desde un cliente, debemos utilizar la función "OpenTCPStream":

```
OpenTCPStream (IP$, Puerto)
```

La dirección IP puede escribirse con la notación "1.2.3.4" o bien con un nombre *www.dominio.com*.

Por ejemplo, queremos abrir una dirección de Internet para enviar una frase al servidor:

```
Puerto_servidor= 60
Servidor = OpenTCPStream
("www.dominio.com",Puerto_servidor)
WriteString Servidor, "Mi nombre
es Darío"
CloseTCPStream Servidor
```

TCP / IP
Información ordenada
Fiable
Lento
Internet
Juegos por turnos

5

Características de una comunicación por TCP / IP.

Al igual que ocurre con los ficheros es necesario volver a cerrar un "Stream" abierto mediante la función "CloseTCPStream".

Un servidor puede estar preparado para recibir cualquier "Stream" de un cliente. Por ejemplo, en el ejemplo anterior, para que el servidor imprima la frase "Mi nombre es Darío" que el cliente le envió debemos utilizar la función "AcceptTCPStream":

```
Servidor = CreateTCPServer (60)
Repeat
 Envio = AcceptTCPStream (Servidor)
Until Envio
ReadString (Envio)
Print Envio
```

En ocasiones puede ocurrir que no se conozca la dirección del cliente de un determinado "Stream" o el puerto de comunicación. Para averiguar ambos datos disponemos de la función "TCPStreamIP" para el IP y "TCPStreamPort" para el puerto:

```
  TCPStreamIP (Manipulador del
  Stream TCP)
  TCPStreamPort (Manipulador del
  Stream TCP)
```

La comunicación TCP suele ser realmente lenta en ocasiones, por ello Blitz3D también permite controlar el tiempo que tarda un "Stream" de datos en llegar. Por defecto, el tiempo de lectura de datos está establecido en 10 segundos, por lo que si un "Stream" tarda más de ese tiempo en llegar ocurrirá un error de transferencia y el "Stream" dejará de estar disponible para su uso. Para variar este tiempo establecido tenemos la función "TCPTimeouts":

```
  TCPTimeouts  Tiempo de espera
  de lectura en milisegundos
  Tiempo de espera en aceptar
  una conexión en milisegundos
```

UDP
Datagramas
Impreciso
Rápido
Red local
Multijugador
- Mas de 2 jugadores
- Tiempo real

6

Características de una comunicación por TCP / IP.

El segundo parámetro sirve para modificar el comportamiento de la función "AcceptTCPStream" para controlar el tiempo que ésta debe esperar para una conexión. Por defecto, el valor es 0, lo que indica que si no hay conexión, "AcceptTCPStream" dejará de trabajar inmediatamente.

FUNCIONES PARA GESTIONAR EL PROTOCOLO UDP

Este tipo de protocolo de comunicación se utiliza para la transmisión de datagramas. Por lo tanto, los paquetes de información que se envían llegan en distinto orden porque

DEFINICIÓN

▶ DATAGRAMA

Un datagrama es una manera de dirigir los paquetes de información por la red local hacia su destino, independientemente unos de otros y utilizando el camino más rápido.

NOTA

Para realizar un juego multijugador donde participan más de 2 jugadores es necesario utilizar comunicación por UDP.

PC Remoto

```
Puerto = 80 ; Puerto del Servidor
Servidor =
OpenTCPStream ("www.d.com",Puerto)
WriteString Servidor, "Hola"
CloseTCPStream Servidor
```

PC Servidor

```
Servidor =CreateTCPServer (80)
Repeat
   Envio = AcceptTCPStream (Servidor)
Until Envio
ReadString (Envio) : Print Envio
```

7

Ejemplo de comunicación simple utilizando funciones de manejo del protocolo TCP / IP.

PC Remoto

```
StreamRem =CreateUDPStream (50)
WriteByte StreamRem, Tipo_mensaje
WriteString StreamRem, "Hola"
SendUDPMsg StreamRem, IPServ, 52
```

PC Servidor

```
StreamServ =CreateUDPStream (52)
Datos=RecvUDPMsg (StreamServ)
Tipo_mensaje = ReadByte (StreamServ)
Saludo$ = ReadString (StreamServ)
```

8

Ejemplo de comunicación simple entre dos ordenadores utilizando funciones de manejo UDP.

cada uno sigue un camino diferente. Por ello, no es un protocolo usado para Internet, pero sí en sistemas multijugador por red local.

Con Blitz3D se puede perfectamente simular las facilidades que nos brindan el uso de las DirectPlay para la realización de un sistema multijugador utilizando funciones UDP. Generalmente, se suele mezclar con funciones TCP para lograr mayor efectividad, sobre todo en juegos con posibilidad de chat.

Para entender mejor las funciones UDP es preciso conocer algunos conceptos de cómo organizar la información en los "Stream". Ya que los paquetes de información a través del protocolo UDP vienen sin acuse de recibo debemos proporcionarlo nosotros mismos. Generalmente, la mejor manera es encabezar cada paquete con un número que indique el tipo de datos que contiene. El orden tiene que ser el mismo en el envío y en la recepción. Todas las transmisiones se realizan sobre un mismo "Stream" por lo que solamente necesitamos crearlo al principio del programa. A través de él viajarán todos los paquetes de datos. Para crearlo utilizamos la función "CreateUDPSteam":

```
CreateUDPStream
([Puerto de
comunicación])
StreamServidor =
CreateUDPStream()
```

El puerto es opcional. Si no se define ninguno la comunicación se realizará a través del primero que se encuentre libre. Si esto ocurre, el cliente podrá saber cuál se utilizó por medio de "UDPStreamPort":

```
UDPStreamPort
(nombre del Stream)
```

Cuando se haya terminado con el proceso de comunicación (generalmente al finalizar el programa) es necesario cerrar el "Stream" con "CloseUDPStream":

```
CloseUDPStream  nombre del Stream
```

Para enviar un paquete de datos tenemos la función "SendUDPMsg". Pero antes debemos crearlo con las funciones normales de escritura de "Stream": "WriteByte", "WriteShort", "WriteInt", "WriteFloat", "WriteString", "WriteLine":

```
SendUDPMsg Stream, IP del
destinatario [,Puerto del destino]
```

Por ejemplo, deseamos mandar las coordenadas X e Y de nuestra nave al servidor de la partida, del cual ya sabemos su IP y su puerto:

```
...
Stream_Remoto = CreateUDPStream
(2001)
...
Tipo_mensaje = 2
WriteByte Stream_Remoto Tipo_mensaje
WriteInt Stream_Remoto   X
WriteInt Stream_Remoto   Y
SendUDPMsg Stream_Remoto
IP_Servidor  Puerto_Servidor
```

Como se puede comprobar en este ejemplo, el "Stream" se va cargando secuencialmente con los datos para ser enviado posteriormente.

En el otro extremo de esta comunicación se deben leer de la misma manera estos datos, recibiendo el "Stream" enviado por el cliente en el "Stream"creado por el servidor con la función "RecvUDPMsg" de la siguiente manera:

```
...
Stream_Servidor =
CreateUDPStream (2000)
...
Datos = RecvUDPMsg
(Stream_Servidor)
Tipo_mensaje = ReadByte
(Stream_Servidor)
X = ReadInt (Stream_Servidor)
Y = ReadInt (Stream_Servidor)
...
```

Todavía nos queda aprender el funcionamiento de otros comandos que manejan la comunicación UDP, pero lo dejamos para la próxima entrega.

En el próximo número...

... terminaremos de explicar el protocolo UDP y empezaremos con los sistemas multijugador usando funciones para el manejo de las DirectPlay.

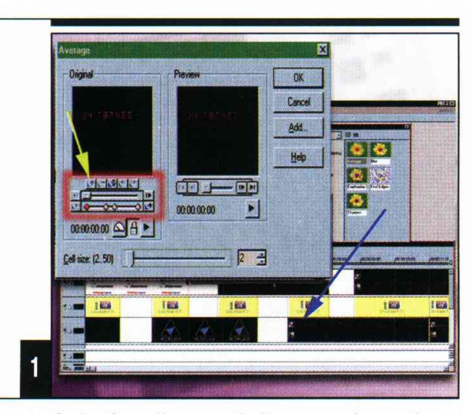
Realización de un vídeo para nuestro juego (III)

Continuando con nuestro vídeo de presentación del juego, terminaremos en esta entrega el montaje del apartado visual con el editor de Ulead MediaStudio.

Veremos la manera de utilizar los filtros de vídeo y las trayectorias de movimiento.

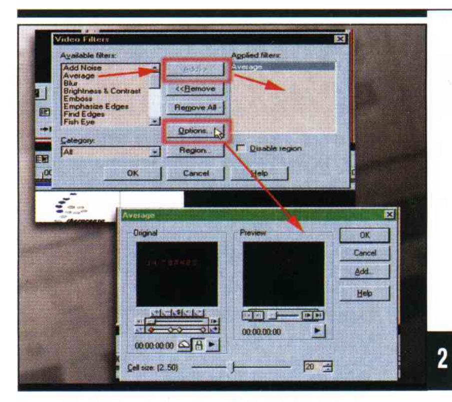
EDITANDO LOS FILTROS DE VÍDEO

Para asignar un filtro de vídeo a un clip solo tenemos que elegirlo de la librería y con el ratón arrastrarlo hacia el clip de la misma manera que aplicamos cualquier efecto de transición. Disponemos de más de 20 filtros diferentes que proporcionan multitud de efectos en la película como: procesos de color, mapeado de texturas, cambios de enfoque, etc. Como comentábamos en el número anterior, podemos aplicar cuantos filtros de vídeo queramos a un clip. Al arrastrar el filtro elegido hacia el clip se abrirá una ventana de opciones en la que podemos cambiar sus características. Cada uno de ellos dispone de distintos parámetros, los cuales aparecerán en la ventana de opciones junto al sistema estándar de visualizado de previo y el keyframer.



Cada efecto lleva asociado sus propios parámetros pero el keyframer y su funcionamiento es el mismo para todos.

Para nuestro primer clip (la primera frase que se fundirá desde negro después del logotipo del desarrollador) aplicaremos un efecto de enfoque y desenfoque mediante el filtro *Average* situado en la carpeta *Video Filter / Focus* de la librería de producción *Production Library*. Para efectuar el cambio de enfoque durante la duración del clip es preciso crear varios puntos o keyframes en el keyframer. Nos situamos en el primer punto del keyframer haciendo clic sobre él. En este lugar del clip aplicaremos un desenfoque del 16 o 20 % desplazando el deslizador *Cell Size* hasta dicho valor. Seguidamente, enfocaremos la imagen durante unos instantes en la mitad de la duración del clip. Así que debemos crear dos puntos más entre los puntos inicial y final del keyframer. Llevamos el deslizador de éste hasta la mitad y pulsamos sobre el pequeño botón con el signo "+" para sumar un nuevo punto, luego llevamos la cantidad de desenfoque (*Cell Size*) hasta el 0%. Hacemos lo mismo con otro nuevo punto que situamos un poco más adelante en el keyframer. Hasta ahora, la imagen del clip pasará de un desenfoque total en el primer punto hasta un enfoque completo hasta el segundo punto, lo cual se mantendrá hasta el tercer punto creado. Para finalizar, hacemos clic sobre el último punto del keyframer y volvemos a aplicar un 20 % de desenfoque. Si pulsamos en cualquiera de los dos iconos de "play" podemos ver el resultado en la ventana de *preview* (Fig. 1).



Podemos aplicar a cada clip cuantos filtros de vídeo queramos.

Otra manera de aplicar o editar filtros de vídeo o trayectorias de movimiento sería a través del menú desplegable que aparece al hacer clic con el botón derecho del ratón sobre el clip. Entonces elegimos las opciones *Video Filters* o *Moving Path*. Si elegimos los filtros aparecerá una ventana de diálogo donde encontramos una lista de los filtros disponibles (*Available filtres*), los cuales podemos pasar a la lista de filtros aplicados (*Applied filtres*) para su utilización pulsando sobre el botón "Add>". Para eliminar un filtro elegido solo tenemos que seleccionarlo de la lista y pulsar sobre "<<Remove". Para editar

NOTA

Cada filtro de vídeo tiene sus propios parámetros, los cuales aparecerán en la ventana de opciones. Los cambios de una modificación de estos parámetros a otra se harán a través de los keyframes sumados en el keyframer.

3

Pulsando en el icono de la lupa podemos ampliar la ventana Motion Control.

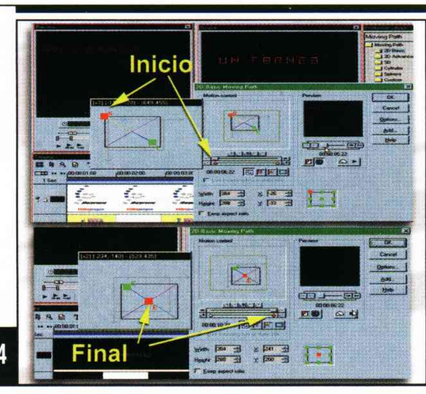
un filtro, lo seleccionamos de la lista y pulsamos sobre el botón "Options" (Fig. 2).

EDITANDO TRAYECTORIAS DE MOVIMIENTO

El segundo efecto que aplicamos al clip será un pequeño desplazamiento por la pantalla de la imagen. Evidentemente, este desplazamiento se sumará al efecto de enfoque y desenfoque anteriormente aplicado y al efecto de transición entre canales. Solo podemos tener una trayectoria diferente por clip. Elegimos de la librería de producción la trayectoria *2D Basic* de la carpeta *Moving Path / 2D Basic* y arrastramos el efecto hacia el clip. Aparecerá una ventana de diálogo con los parámetros del efecto. En esta ocasión nos encontramos con una cantidad superior de opciones, realmente muy sencillas de utilizar. Lo primero que tenemos que saber es que toda la trayectoria del efecto se trabaja en *Motion control,* situado a la izquierda de la ventana. Antes de empezar a trabajar, ampliemos esta ventana pulsando sobre el icono de la lupa. Se abrirá una nueva ventana con un modelo ampliado del *Motion control*. Disponemos de dos niveles de aumento haciendo clic con el botón izquierdo sobre la ventana para aumentar y con el derecho para disminuir. Podemos elegir la zona aumentada desplazando con el ratón el cuadrado de color verde situado en el *Motion control* y que indica la zona la totalidad de la pantalla. El cuadradito con la letra "S" indica el primer punto del keyframer y el de la letra "E" el último. Si añadimos un nuevo punto aparecerá un nuevo cuadradito pero sin letra. Al hacer clic sobre cualquier punto, su cuadrado correspondiente se volverá de color rojo, indicando que podemos desplazarlo con el ratón. Si pulsamos el icono con un pequeño monitor situado debajo de la ventana *Preview*, extenderemos la visualización del efecto a la ventana *Preview* principal del programa (Fig. 3).

NOTA

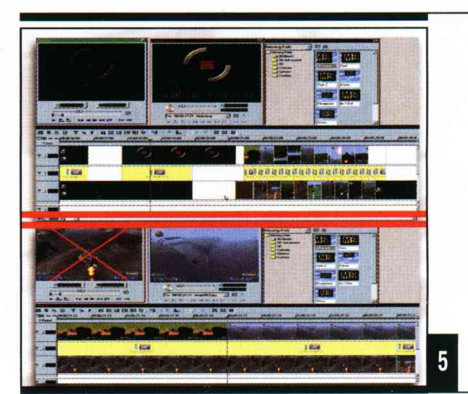
Podemos borrar los filtros de vídeo o las trayectorias de un clip eligiendo la opción *Delete Attributes* del menú desplegable que aparece al hacer clic con el botón derecho sobre el clip.



La ventana Motion Control sirve para configurar el movimiento del clip en la pantalla para cada filtro de trayectoria.

Ya estamos preparados para editar nuestra trayectoria. Hacemos clic sobre el cuadradito con la "S" o sobre el primer punto y lo desplazamos hacia el extremo superior izquierda de la pantalla (podemos ver el resultado inmediatamente en la ventana de previo). Luego, hacemos clic en el cuadrado con la "E" y lo desplazamos hasta que el extremo inferior derecho. La idea es que la palabra "UN TORNEO" cruce parcialmente la pantalla de esquina a esquina, de tal forma que el momento de enfoque completo coincida con el centro de la pantalla (Fig. 4).

El resto de imágenes con texto se añade aplicando el mismo procedimiento. Es preciso aplicar trayectorias con diferentes direcciones para conseguir un cierto orden en el visualizado de los textos.



Para conseguir visualizar un mosaico de imágenes de medio segundo de forma suave un buen truco es mezclarlas con un efecto Crossfade.

APLICANDO EL RESTO DE IMÁGENES

Para finalizar el vídeo visualizaremos un mosaico con capturas del juego a gran velocidad. Para suavizar el visionado mezclamos cada captura con un efecto de transición *Crossfade*. La duración de cada imagen será de medio segundo y el efecto de transición englobará la mitad de ese medio segundo. De esta forma conseguimos una mayor continuidad (Fig. 5).

Para finalizar, rompemos la secuencia de imágenes del juego con un clip de color degradado para fundir con el último texto en movimiento.

Terminado el montaje del vídeo nos queda aplicar el audio a la película y guardar el resultado en un fichero de vídeo.



En el próximo número...

... aplicaremos el audio a nuestro vídeo y añadiremos algunos efectos con el programa Illusion.

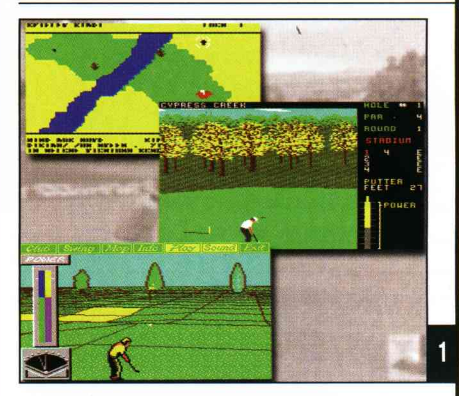
Simuladores deportivos (y II)

Con esta entrega, completamos la historia de los juegos deportivos más significativos para ordenador con los simuladores de golf, tenis y el subgénero "extreme".

En la próxima entrega comenzaremos con un nuevo género: los simuladores de velocidad.

SIMULADORES DEPORTIVOS (II)

La historia de los juegos de golf para ordenador es relativamente joven. No es un género deportivo mayoritario y profesionalmente las producciones se reducen a un círculo más o menos cerrado de títulos. Prácticamente, fue en 1982 cuando se escribió el primer juego de golf para ordenador por Bruce Caver, concretamente para Commodore 64, y se llamaba *World Class Leaderboard Golf*. Después de publicar este juego, Caver crea Access Software para seguir desarrollando este tipo de juegos. Este título fue el antecesor de la serie más popular del género, conocida como *Links*. Access se convirtió en la cuna de los desarrolladores de juegos de golf como Vance Cook, que formó su propia compañía en 1992, la cual se encargó de desarrollar la serie *Tiger Woods* para EA. Volviendo atrás, antes de la década de los 90 aparece una de las series con más títulos en el mercado con cerca de una treintena de juegos de golf para todas las plataformas, *Jack Nicklauss* (Accolade, Inc). Para PC surge por primera vez en 1988 *Jack Nicklaus´ Greatest 18 Holes of Major Championship Golf*, el juego permitía jugar hasta 18 hoyos con el ordenador o con 8 jugadores ficticios. En el curso de los hoyos se podían encontrar multitud de árboles, bunkers e incluso agua. Se podía controlar la fuerza del golpe con una barra de potencia además de disponer de toda clase de estadísticas del partido. En los comienzos, generalmente se utilizaba la vista cenital sobre una pantalla fija que representaba partes del hoyo en el transcurso de la partida. La serie *Jack Nicklauss* introduce como norma una representación similar a la tercera persona en donde la cámara se sitúa detrás del jugador, el cual lanzaba la pelota hacia delante. Con la serie *Links*, Access introduce de forma magistral los entornos en 3D predibujados con terrenos simulando desniveles en 3D. *Links* se extiende a lo largo de más de una década llena de títulos que siguen el rumbo de la tecnología. Ya en 1999 se toma como norma la captura de movimientos para los jugadores y poco a poco se sustituye el curso de los hoyos fotorrealistas por entornos completamente en 3D donde es posible verlos desde cualquier posición. También, por ejemplo, en *Links 2001* se implementan barrancos y trampas de arena, mejorando la física de la pelota para que se adapte a un terreno más complejo. Además se sustituye el teclado por el uso exclusivo del ratón para realizar los golpes. Por otro lado, Microsoft compite con Access con la serie *Microsoft Golf*, realizando títulos prácticamente idénticos a la serie *Links*. Aun así, Microsoft adquiere los derechos de *Links* y compra Access. Básicamente, la línea de las desarrolladoras de este género se centra en buscar licencias de jugadores de elite como Tiger Woods o Sergio García, buscando siempre torneos de renombre y simulando a la perfección los

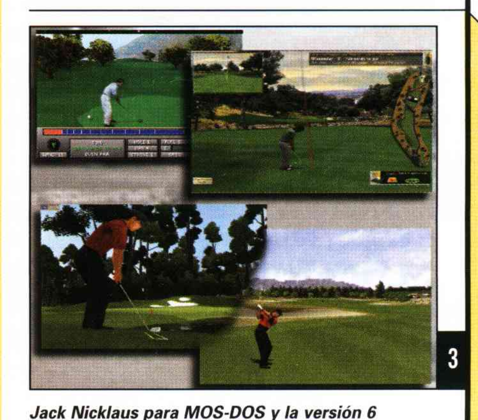


1

La perspectiva en tercera persona fue impuesta por juegos como World Class LeaderBoard Glof (Centro) o Greg Norman´s Ultimate Golf (abajo).


2

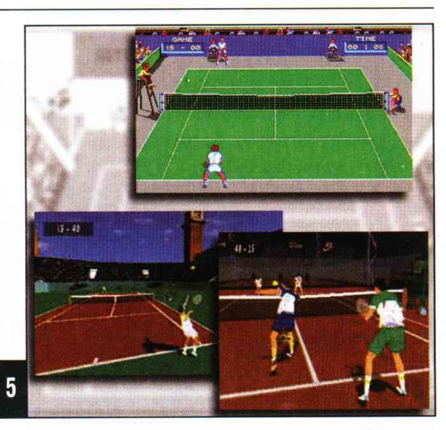
La serie Links de Access siempre tan brillante incluso para MS-DOS. Links 386 Pro (DOS, arriba), Links 2001 (abajo).


3

Jack Nicklaus para MOS-DOS y la versión 6 (arriba). El movimiento de la cámara permite ver el golpe desde cualquier ángulo.

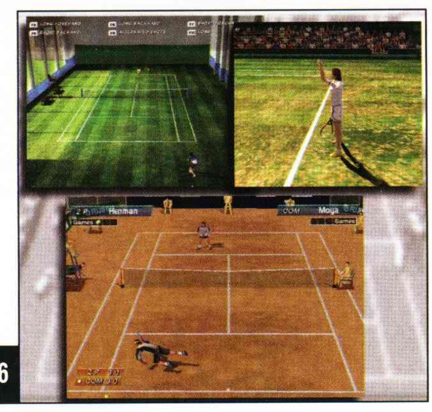
4

Pong (arriba), el primer juego de tenis y el sencillo e increíblemente jugable Match Point de Psion (abajo).



5

Dos clásicos de consola versionados para PC. Great Courts Tennis (arriba) y Pete Sampras Tennis 97 (abajo).



6

El increíble realismo alcanzado por los juegos de tenis actuales está representado por Roland Garros 2000 (arriba) y Virtual Tennis (abajo).

campos más importantes del mundo con entornos fotorrealistas en 3D como en *Links 2001/02/03*, *Tyger Woods PGA Tours 2001/02/03* o la serie *PGA ChampionShip Golf*.

Otro género deportivo que ha estado presente a lo largo de la historia del videojuego es sin duda el tenis. Como ya sabemos, el primer videojuego electrónico fue el *Pong* y simulaba un partido de tenis. El resto ya lo conocemos. Pero para hablar realmente de la representación de dos jugadores en una cancha golpeando una pelota con raquetas tenemos que irnos un poco más adelante, hasta la época de los 8 bits. Allí encontramos una verdadera joya, *Match Point (Psion)*, quizás el mejor juego de tenis a pesar de su edad. Poseía una sencilla combinación de golpes y unos gráficos simples. El campo se mostraba con vista cenital ligeramente tumbada. Pero nada de esto enturbiaba la endiablada jugabilidad que tenía. Quizás este concepto, la jugabilidad, es lo que persigue este tipo de simuladores. Lejos del detalle, aunque importante, se busca siempre el toque arcade para lograr tan deseada jugabilidad. Y el juego que mejor la desarrollaba fue *Super Tennis* para la consola *SNES*. No hay muchos títulos de este género para PC, la mayoría son versiones de consola. De los primeros, destacar la conversión *Great Courts: The Ultimate Tennis Simulation* (Blue Byte, Ubi Soft, 1989) para MS-DOS con hasta 4 jugadores. O el intento por simular gráficos en 3D de Distinctive Software, Inc con *4D Tennis* en el año 1989.Pero sin lugar a dudas, la mejor conversión se realizó desde la PSX con *Pete Sampras Tennis 97* por Codemasters con un nivel arcade fantástico. Con la llegada de la aceleración gráfica se pasa de simular la perspectiva 3D a modelar los jugadores y el entorno en 3D real con lo que se consigue una física de la pelota más convincente. Además, se incluye la posibilidad multijugador en la serie *Roland Garros*. Actualmente, el líder indiscutible de simuladores de tenis es Sega con su título *Virtual Tennis* para Dreamcast y PS2. Sega, aprovechando el hueco existente en PC, convierte este título para esta plataforma incluyendo el modo multijugador evitando así cualquier competencia. Los jugadores poseen animaciones por captura de movimientos y es posible jugar en cualquier tipo de terreno con un nivel de detalle tal que a veces parece un partido televisado.

Dejando aparte otros géneros deportivos como el billar, el boxeo, etc, hacemos un pequeño repaso a los juegos denominados "Radical Sports" con el mejor representante en la serie *Extreme*. Los juegos extremos empiezan en PC con *Extreme Sports* en 1994 y continúan ofreciendo gran cantidad de modalidades diferentes desde rodeo hasta ciclismo de montaña. La serie más popular y tecnológicamente más avanzada de juegos extremos es sin duda *ESPN Extreme Games* (Sony, 1995), poseía gráficos en 3D, animaciones mediante captura de movimientos, voz y vídeo y una gran diversidad de deportes "callejeros". Pero el deporte extremo con mayor representación en el PC es el "snowboard" y todas sus variantes, en los denominados "juegos de invierno". Pero antes, comentar que los juegos de invierno tuvieron su primer representante digno en el año 1986 con *Winter Games* (Action Graphics) publicado por Epyx (los cuales publicaron un año después su equivalente en juegos de "verano" con la serie *California Games*), con más de diez pruebas diferentes. De los primeros juegos multijugador de "snowboard" destacamos *ESPN X Games Pro Boarder* (Radical Entertainment, 1998) y *Boarder Zone 2000* con gráficos en 3D.



En el próximo número...

... veremos los simuladores de velocidad.

Cuestionario Videojuegos



Preguntas

1. ¿Cuáles son los protocolos de comunicación utilizados por Blitz3D para conectar ordenadores entre sí y en qué se diferencian?
2. Escribe un código en Blitz3D para enviar un mensaje a otro ordenador utilizando comunicación TCP y lo mismo pero para UDP.
3. ¿Cómo podemos cambiar (en Blitz3D) el panorámico de un sonido a partir de la posición de la entidad emisora?
4. Escribe el código en Blitz3D para mantener una música en reproducción continua durante la partida.
5. ¿Cómo podemos obtener de una manera rápida un estilo de polígono en Paint Shop Pro?
6. ¿Qué debemos tener en cuenta a la hora de diseñar puntos de mira en un juego en primera persona?
7. Enumera los formatos de canción que utiliza para trabajar ModPlug Tracker.
8. ¿Qué elementos forma cada línea de un patrón en ModPlug Tracker?
9. ¿Cómo podemos aplicar más de un filtro de vídeo a un clip en MediaStudio sin usar la librería de producción directamente?
10. En el editor de vídeo de MediaStudio, ¿dónde modificamos los parámetros de un filtro de trayectoria?

Respuestas al cuestionario 17

▷ **1.**

```
sonido1=LoadSound("s1.wav"): sonido2=LoadSound("s2.wav")
Canal_S1=PlaySound(sonido1): Canal_S2=PlaySound(sonido2)
Volume#=1.0
While Not Keyhit(1)
    If Not ChannelPlaying (Canal_S1)
        Canal_S1=PlaySound(sonido1)
        Canal_S2=PlaySound(sonido2)
    Endif
        If KeyDown (200) Volumen#=Volumen+.1
        If KeyDown (208) Volumen#=Volumen-.1
    ChannelVolume Canal_S1, Volumen#
Wend
FreeSound sonido1: FreeSound sonido2
```

▷ **2.**

```
Graphics 800,600
SetBuffer BackBuffer()
Pelicula=OpenMovie("peli.mpg")
While Not KeyHit(1)
Cls
DrawMovie Pelicula, 0,0,400,300:  DrawMovie Pelicula, 400,0,800,300
DrawMovie Pelicula, 0,300,400,600: DrawMovie Pelicula, 400,3000,400,600
If Not MoviePlaying(Pelicula)
        CloseMovie Película
        Pelicula=OpenMovie("peli.mpg")
    EndIf
    Flip
Wend
CloseFile (Fichero)
```

▷ **3.**

```
Contador = Rand(1000,5000): Contador_Total = Contador
Xdestino#=Rand(0,lado_Terreno):  Zdestino#=Rand(0,lado_Terreno)
Repeat
    Contador=Contador-1
    If Contador = Contador_Total / 2
        Xdestino#=Rand(0,lado_Terreno)
        Zdestino#=Rand(0,lado_Terreno)
    Endif
    If  X#<Xdestino X=X+Desplazamiento
    If  Z#<Zdestino Z=Z+Desplazamiento
    If  X#>Xdestino X=X-Desplazamiento
    If  Z#>Zdestino Z=Z-Desplazamiento
    PositionEntity Objeto, X#, Altura_Terreno#, Z#
Forever
```

▷ **4.** Si las entidades se desplazan: buscar la posición del enemigo, atacar eligiendo arma según la distancia al contrario, cambiar de posición en caso de ataque o cambiar la posición en busca de extras.

▷ **5.** La primera textura generalmente tiene el tamaño del mapa de alturas y la segunda se aplica con tileado.

▷ **6.** Que los bordes de la textura coincidan cuando estén unidos.

▷ **7.** En *Piano Roll* seleccionamos las notas que queremos que se repitan. Seguidamente, elegimos la opción *Repeat selected notes in a Loop* del menú *Edit* y la posterior opción *Just play loop once*.

▷ **8.** El "Performer" es una herramienta de Anvil Studio para crear programas que automaticen la reproducción de sonidos con efectos, cambios de tonos, etc.

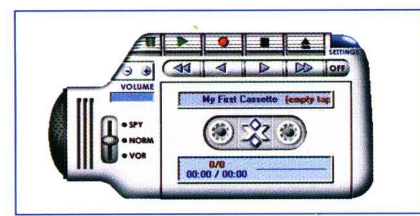
▷ **9.** Podemos mezclar dos pistas o canales de vídeo.

▷ **10.** Arrastrando con el ratón el efecto "Blur" o "Average" desde la librería de producción hacia el clip.

Contenido CD-ROM 18

▶ AUDIO

■ Audio Notes Recorder 3.5

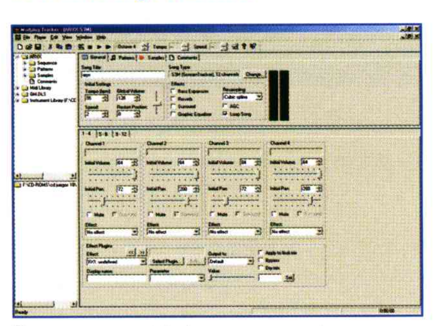


Grabación de voz que te será muy útil para el desarrollo del juego.

■ MP3 Gain

Interesante utilidad para homogeneizar tus archivos de audio.

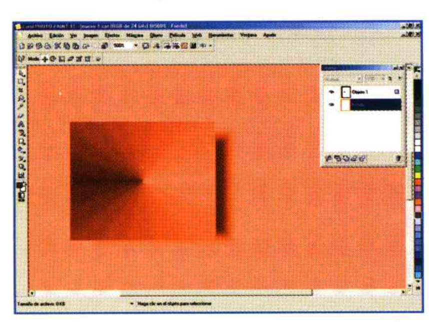
■ ModPlug Tracker



Nueva oportunidad para conseguir este potente tracker con el que desarrollaremos el audio en "Zone of Fighters".

▶ DISEÑO 2D

■ Corel Graphic Suite 11



Magnífica suite de dibujo y retoques con la que podrás realizar auténticas virguerías.

■ PhotoAssist 1.2

Maneja tus imágenes rápida y ágilmente con la ayuda de esta aplicación.

■ PhotoMail 4.0

Crea álbumes con tus imágenes y envíalas por mail a quien tú desees.

▶ DISEÑO 3D

■ CrazyTalk 3.0



Simpática aplicación con la que podrás animar los labios de tus personajes para que hablen.

■ Cosmos 4D

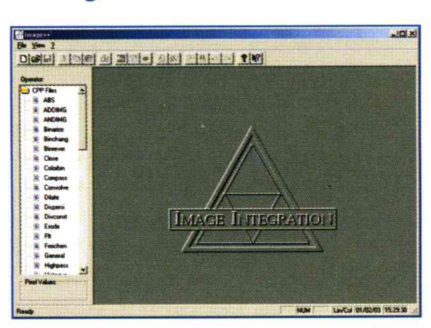
Rápido programa para crear imágenes en varias dimensiones y editarlas.

■ Moray 3.3

Uno de los mejores editor es de pov-ray con el que realizar potentes entornos.

▶ PROGRAMACIÓN

■ Image ++ 2.1



Desarrollo gráfico mediante programación orientada a objetos.

■ Morfit 3D SDK 4.0 para C++



Completo programa de desarrollo 3D bajo C++.

■ Code-Lock 2.10

Bloquea el software que crees para que solo pueda ser utilizado por los usuarios registrados.

■ Hickwall Debugger 1.0

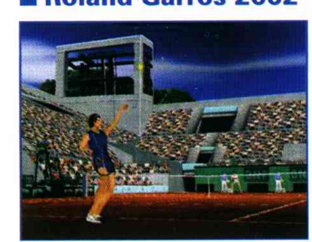
Controla cómo los programas deben ser ejecutados, para que corran tal y como desees.

▶ JUEGOS

■ PGA Championship Golf 2000

Si te entusiasma el golf, con este juego podrás practicarlo virtualmente.

■ Roland Garros 2002



Excelente simulador de tenis con un montón de opciones para jugar.

■ SkyRacer

Deslízate muy rápido por la nieve con tu vehículo y compite en la carrera de este deporte extremo.

■ Zone of Fighters

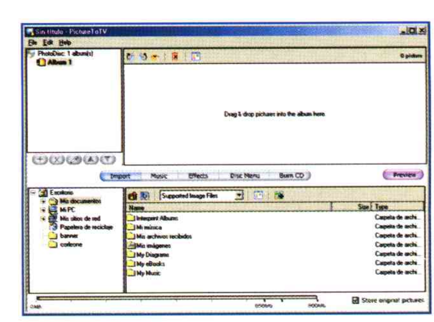
Como todas las semanas, nuestro juego.

▶ VÍDEO

■ DVD Cutter Stream

Interesante utilidad que te permitirá reproducir una gran variedad de formatos de vídeo.

■ PictureToTv



Crea vídeos a partir de imágenes estáticas muy fácilmente usando este programa.

▶ EXTRAS

En este apartado encontrarás todos los ejemplos de los que hablamos en el coleccionable, para que no pierdas detalle.